(12)

User Modeling and Register Theory: A congruence of concerns

Cecile L. Paris and John A. Bateman
USC Information Sciences Institute
4676 Admiralty Way
Marina del Rey, California 90292

November 1990
ISI/RR-93-315

REPORT DOCUMENTATION PAGE

FORM APPROVED
OMB NO. 0704-0188

1. AGENCY USE ONLY *(Leave blank)*

2. REPORT DATE
November 1990

3. REPORT TYPE AND DATES COVERED
Research Report

4. TITLE AND SUBTITLE
User Modeling and Register Theory: A congruence of concerns

5. FUNDING NUMBERS
NCC2-520
F49620-87-C-0005
IRI-9003087
MDA 903-87-C-641

6. AUTHOR(S)
Cecile L. Paris and John A. Bateman

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
USC INFORMATION SCIENCES INSTITUTE
4676 ADMIRALTY WAY
MARINA DEL REY, CA 90292-6695

8. PERFORMING ORGANIZATION REPORT NUMBER
RR-315

9. SPONSORING/MONITORING AGENCY NAMES(S) AND ADDRESS(ES)

NASA AMES	ARPA	NSF
Moffett Field, CA 94035	3701 N. Fairfax Drive	1800 G St. NW
	Arlington, CA 22203	Washington, DC 20550

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES

12A. DISTRIBUTION/AVAILABILITY STATEMENT
UNCLASSIFIED/UNLIMITED

12B. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*

Sophisticated computer systems using natural language to interact with people are now becoming widespread. These systems need to communicate with an increasingly varied user community, across an ever more extensive range of situations. Just as for human-human interaction, no single style of generated text is adequate across all user types and all situations. Generation systems can only be effective if they appropriately 'taylor' their phrasing, text content, and organization according to the situation and to the abilities and requirements of the intended readers. This paper presents new work in 'tailoring' that addresses the phrasing problem: how to best express the propositional content that has been chosen by a text planner, given a user and situation. Importantly, this paper shows how relevant linguistic studies can be bought to bear the problem of user modeling and tailoring. In particular, we would like to show that the concerns of register theory are very close to some of the concerns of user modeling, and that aspects of the theory can guide us in studies in user modeling. Based on this specific linguistic theory, we propose a methodology to systematically study the problem of tailoring phrasing.

14. SUBJECT TERMS
tailoring, user modeling, register theory, generation systems

15. NUMBER OF PAGES
33

16. PRICE CODE

17. SECURITY CLASSIFICTION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
UNCLASSIFIED	UNCLASSIFIED	UNCLASSIFIED	UNLIMITED

GENERAL INSTRUCTIONS FOR COMPLETING SF 298

The Report Documentation Page (RDP) is used in announcing and cataloging reoprts. It is important that this information be consistent with the rest of the report, particularly the cover and title page. Instructions for filling in each block of the form follow. It is important to stay within the lines to meet optical scanning requirements.

Block 1. Agency Use Only (Leave blank).

Block 2. Report Date. Full publication date including day, month,a nd year, if available (e.g. 1 jan 88). Must cite at least the year.

Block 3. Type of Report and Dates Covered. State whether report is interim, final, etc. If applicable, enter inclusive report dates (e.g. 10 Jun 87 - 30 Jun 88).

Block 4. Title and Subtitle. A title is taken from the part of the report that provides the most meaningful and complete information. When a report is prepared in more than one volume, repeat the primary title, add volume number, and include subtitle for the specific volume. On classified documents enter the title classification in parentheses.

Block 5. Funding Numbers. To include contract and grant numbers; may include program element numbers(s), project number(s), task number(s), and work unit number(s). Use the following labels:

C	- Contract	PR	- Project
G	- Grant	TA	- Task
PE	- Program Element	WU	- Work Unit Accession No.

Block 6. Author(s). Name(s) of person(s) responsible for writing the report, performing the research, or credited with the content of the report. If editor or compiler, this should follow the name(s).

Block 7. Performing Organization Name(s) and Address(es). Self-explanatory.

Block 8. Performing Organization Report Number. Enter the unique alphanumeric report number(s) assigned by the organization performing the repor.

Block 9. Sponsoring/Monitoring Agency Names(s) and Address(es). Self-explanatory

Block 10. Sponsoring/Monitoring Agency Report Number. (If known)

Block 11. Supplementary Notes. Enter information not included elsewhere such as: Prepared in cooperation with...; Trans. of ...; To be published in... When a report is revised, include a statement whether the new report supersedes or supplements the older report.

Block 12a. Distribution/Availability Statement. Denotes public availability or limitations. Cite any availability to the public. Enter additional limitations or special markings in all capitals (e.g. NOFORN, REL, ITAR).

DOD	- See DoDD 5230.24, "Distribution Statements on Technical Documents."
DOE	- See authorities.
NASA	- See Handbook NHB 2200.2.
NTIS	- Leave blank.

Block 12b. Distribution Code.

DOD	- Leave blank.
DOE	- Enter DOE distribution categories from the Standard Distribution for Unclassified Scientific and Technical Reports.
NASA	- Leave blank.
NTIS	- Leave blank.

Block 13. Abstract. Include a brief (Maximum 200 words) factual summary of the most significant information contained in the report.

Block 14. Subject Terms. Keywords or phrases identifying major subjects in the report.

Block 15. Number of Pages. Enter the total number of pages.

Block 16. Price Code. Enter appropriate price code (NTIS only).

Blocks 17.-19. Security Classifications. Self-explanatory. Enter U.S. Security Classification in accordance with U.S. Security Regulations (i.e., UNCLASSIFIED). If form contins classified information, stamp classification on the top and bottom of the page.

Block 20. Limitation of Abstract. This block must be completed to assign a limitation to the abstract. Enter either UL (unlimited) or SAR (same as report). An entry in this block is necessary if the abstract is to be limited. If blank, the abstract is assumed to be unlimited.

User Modeling and Register Theory: A congruence of concerns

Cécile L. Paris and John A. Bateman
USC/Information Sciences Institute
4676 Admiralty Way
Marina del Rey, CA 90292-6695
U.S.A.

November 1990

Abstract

Sophisticated computer systems using natural language to interact with people are now becoming widespread. These systems need to communicate with an increasingly varied user community, across an ever more extensive range of situations. Just as for human-human interaction, no single style of generated text is adequate across all user types and all situations. Generation systems can only be effective if they appropriately 'tailor' their phrasing, text content, and organization according to the situation and to the abilities and requirements of the intended readers. This paper presents new work in 'tailoring' that addresses the *phrasing problem*: how to best express the propositional content that has been chosen by a text planner, given a user and a situation. Importantly, this paper shows how relevant linguistic studies can be bought to bear on the problem of user modeling and tailoring. In particular, we would like to show that the concerns of register theory are very close to some of the concerns of user modeling, and that aspects of the theory can guide us in our studies in user modeling. Based on this specific linguistic theory, we propose a methodology to systematically study the problem of tailoring phrasing.

1 Introduction

Sophisticated computer systems using natural language to interact with people are now becoming widespread. These systems need to communicate with an increasingly varied user community, across an ever more extensive range of situations. Just as for human-human interaction, no single style of generated text is adequate across all user types and all situations. Generation systems can only be effective if they

appropriately 'tailor' their phrasing, text content, and organization according to the situation and to the abilities and requirements of the intended readers.

In general, tailoring language to a situation requires an *explicit* account of how textual, lexical and grammatical linguistic resources may be deployed depending on their context of use. The realization that situations, which *include* user types, *systematically* affect language can greatly help in providing this account. Not only do people talk differently in different situations, but they do so predictably. Specific types of situations have definable properties, which in turn have determinate consequences on language. That is: given appropriate classifications of situation-types, the consequences of these situations on language can be specified, and suitable language can be produced. This is exactly the premise of *register theory* (e.g., [11, 19]), an aspect of Systemic-Functional Linguistics that is precisely concerned with how language varies according to the situation. In this paper, we would like to show that aspects of register theory can guide us in our studies in user modeling.

Situation-dependent linguistic variation permeates all levels of linguistic realization, from the content and organization of the text as a whole to finer decisions as to which aspects of the information to express, and which lexical and syntactic constructions should be chosen for individual sentences (i.e., the *phrasing* of the text). Much previous research on text generation and user modeling has focused on building user models to guide the selection and organization of the appropriate information from a knowledge base for presentation to the user. But this is not enough. It is also important that the phrasing be tailored to the user. Otherwise, generated texts may be just as ineffective as texts that misdirect attention or rely on knowledge the reader does not have.

As a step toward the development of a framework within which tailoring to the situation can occur at *all* levels of linguistic realization, the research we describe here is aimed specifically at tailoring the *phrasing* of the sentences of a text according to the situation. In a pilot study that involved generating explanations for an expert system tailored to three types of users [2], we showed that it is possible to specify the type of language required in a given situation and to build a text planning system that uses that specification to vary the phrasing of its text. In that study, we restricted our attention to a few kinds of texts in the domain of digital circuit diagnosis. We analyzed these texts and the situations in which they appeared. Based on our analysis, we constructed specifications of the important features of the situation, and of the linguistic variations among the texts. The situational features were then used as an additional source of input to the text generation process. They controlled the grammar, thus controlling linguistic variation.

We are now furthering this work to create a framework within which it is possible to gain systematic control over phrasing. Based on the results and methodology of

specific aspects of a linguistic theory, we study how language varies depending on the situation, and show how a system can systematically gain control over the phrasing process.

This research has implications for the fields of both user modeling and text generation:

- for user modeling, we are uncovering dimensions and features that a user model needs to contain in order to control a grammar effectively; moreover, register theory, the linguistic theory used as the theoretical framework for this work, provides guidance in the representation of these features, and a methodology for furthering the understanding of *how* user characteristics affect language;
- for text generation, we are forced to develop further mechanisms to control a grammar and to allow a text planner to control fine grammatical decisions.

In this paper, we first illustrate the 'phrasing problem' and its importance by showing some examples from naturally occurring texts. We then present the theoretical framework that we have adopted in attempting to solve this problem. We briefly describe the current status of the research and show how we were already able to give our text planner a limited ability to control fine-level grammatical details in response to a specific situation. We then describe how the theoretical framework we are adopting will help us in furthering the research. In particular, we describe how it will help us design methodologies and mechanisms by which the systematic relationship between situations and language can be found, represented, and used to tailor generated texts.

2 The problem: natural language variation

People speak differently according to the situation. For example, a surgeon will describe some aspects of a surgical operation differently depending on whether s/he is in a briefing room, actually carrying out the operation and talking with other doctors and nurses, or discussing it afterwards with his/her friends; two doctors reviewing a patient's problem will employ precise and 'technical' medical terms; on the other hand, when talking to the patient, they will use a different style of language in order for the patient to understand them. This is illustrated quite strikingly in the anecdote shown in Figure 1.

Importantly, these differences in language are due to a great extent to the choices *at fine level of detail* of exactly which aspects of a specific propositional content is to

When I was training to become an emergency medical technician, the physician in charge stressed the importance of using proper medical terminology. Soon after my graduation, I had to transport a boy with a head wound to the hospital, so I wrote in the description: 'Ten-year-old male with ten-centimeter laceration on the left occipital region.'
The doctor who had instructed me met us in the emergency room. 'What happened, son?' he asked the child. 'Did you bop your gourd?"

Figure 1: Employing different terms to talk about a medical problem

be expressed or highlighted, and to differences in the lexical and syntactic structures chosen to express the information. That is, these differences in language are due to a great extent to the *phrasing* of the various texts. Thus, if they are to communicate effectively, computational systems must also be capable of appropriately tailoring the phrasing of texts they produce.

This phenomenon can be further illustrated by considering the following examples, in which the situation and the intended readers causes the same subject matter to be presented differently, both in terms of the structure of the text and its syntactic forms (e.g., shorter sentences, use of declarative as opposed to imperative form). Figure 2 contains three possible versions of instructions for updating a file in a computer system, such as might appear in a manual [14]. Each of these texts is more appropriate than another depending on the text's purposes and intended users. The variations are contrasted by Grimm [14] as follows:

- The 'prose style' (2a) is good for overview and introduction but submerges all the instructions in text. As a result, it is not as understandable as (2b) - or (2c) - for the purposes of giving instructions.
- The 'cookbook' style (2b) gives the same instructions as the prose example but brings out much more what needs to be done. It has about 20 fewer words. Shorter reading lines mean faster reading and quicker understanding. This form is particularly good for instructions.
- The 'playscript' style is useful when the instructions are addressed to more than one group of users. The playscript method makes the doer clear. Different users need read only the instructions that apply to them. This form is particularly good for sections of a manual that have multiple users.

It is clear that the intended use or function of a text (overview *vs* instructions, glossary *vs* body of text, number of intended readers) can play a role in the differential phrasing of a text. But this is obviously not the only factor to consider. The sophistication of the users and the users' level of expertise in the domain of the text can

(2a) Prose Style:

Order clerks complete the Purchase Order Input Form (form number PO123). The form has a preprinted purchase order number. The clerk enters the date, supplier company name, the item, the quantity. and the cost.
The clerks send the form to the Data Entry Department where the information is keyed to an input tape. The tape goes to the computer and updates the purchase order file.

(2b) Cookbook Style:

Complete the Purchase Order Form, PO123.
Enter the date, supplier company, item number, quantity, and cost.
Send the form to Data Entry.
Key the data to a tape.
Process the tape on the computer.
Update the purchase order file.

(2c) Playscript Style:

ORDER CLERK 1. Complete the Purchase Order Form (PO123).
(A) PURCHASE ORDER NUMBER The number is preprinted on the form.
(B) DATE Enter the current date. Use 6 digits; the 2-digit month, the 2-digit day, and the last 2 digits of the year. For example, August 6, 1982 is 080682.
(C) SUPPLIER COMPANY Enter the name of the company that will receive the order. Use up to 30 characters.
2. Send the form to Data Entry.

DATA ENTRY PERSONNEL 1. Key the data from form PO123.
2. Give resulting tape to Operations.

OPERATIONS Process the tape to update the purchase order.

Figure 2: Potential instructions for a computer system: variation according to text function

Example text (3a)

As data are entered and processed, they are checked to assure the accuracy of the processing. When an error is detected, the error record will be printed at the terminal (or file if deferred or overnight processing) and the reason for the diagnostic given. In most cases, a special character indicates the "error" portion of the record.

Example text (3b)

The computer will edit data as they are entered. This assures processing accuracy. When the computer finds an error, it will print an error message at the terminal. The error message will explain the reason for the error. (If you selected deferred or overnight processing, the error messages will be on a file.) In most cases, a special character prints under the part of the record that is in error.

Figure 3: Documentation for a computer system: variation according to "ease of readability"

also play a role in determining the syntactic complexity of a text. Figure 3 contains two versions of a section of a computer system manual on error diagnostics [14]. The selection of one of these as more appropriate in a particular situation is sometimes made on the basis of readability tests:[1] (3b) is said to be 'easier' to read than (3a). Therefore, (3b) may be a preferred version if the intended readers were not highly educated.

Figure 4 presents two descriptions of the same medical term, one to a medical expert such as a physician and the other to a layman such as a patient. It is fairly safe to assume that the patient would not understand 'an unpaired compound gland' or 'cordlike extension of the infundibilum'. A system talking to him/her would therefore have to use more common terms and phrases, such as 'pituitary gland'. However, although the expert may well understand the more naive text, it will in many cases be too general or vague to convey accurate information (since one of the principal characteristics of technical language is its detailed accuracy). So a system that addresses an expert must do so in suitably accurate terms. In general, there is no middle ground in this problem, and, therefore, systems that do not tailor their language to the user will not be very useful.

To summarize, examples of actual texts show variation on a number of dimensions simultaneously. The texts show contrasts across text functions (e.g., overviews,

[1] Grimm [14] cites the Gunning Fog Readability Index [17].

Example text (4a); medical dictionary [58]:

hypophysis: Glandula pituitaria or basilaris; pituitary or master gland; h. cerebri; an unpaired compound gland suspended from the base of the hypothalamus by a short cordlike extension of the infundibilum, the hypophysial (or pituitary) stalk.

Example text (4b); ordinary dictionary [63]:

hypophysis: the pituitary gland.

Figure 4: Dictionary entries for doctors and non-doctors

glossaries, instructions), across overall sophistication of the users and across technical expertise (e.g., technical *vs* layman text). There is a need therefore to study the various types of text as well as the various forms in which the text appear (manuals, documents, instructions, etc.). In addition, there is a need to understand in what ways these dimensions of variation interact. Only then will it be possible to produce appropriately tailored texts by computers. Note that a characterization such as that given by Grimm to describe how and when the different styles of text should be produced is not sufficient to guide a text generation system. Similarly, while it is clear that there are indeed differences between the two texts shown in Figure 3, in terms of both the text structures and their grammatical organizations, the complexity measure provided by a readability test does not provide a particularly accurate or revealing sense of those differences and their motivations. In order to achieve systematic control over linguistic variation in a computer system, there is a need to be more explicit about both the differences in situations and the linguistic differences that these situations require. What precisely these differences are, and how to constrain a text generation system's behavior to produce them, are therefore necessary goals for research.

It is of course possible to *handcraft* generation programs to produce each distinct type of phrasing that may be required. This has indeed been done using, for example, the MUMBLE text generation system [42], or the FUG formalism [31], as in TAILOR [49], or other generation system, as in PAULINE [25]. In this research, however, we are seeking a more general solution. We have begun to specify the differences between various situations and various texts and are extending the capabilities of the large-scale, reusable text generation system PENMAN [35] so that it may straightforwardly

generate texts in any particular style. Providing this new capability, however, can be done in an extensible and general way only if it is possible to uncover similarities among distinct situations, user types, and language. These similarities would then provide a basis for stating systematic variations in the language for those users and situations. We propose here a methodology to further develop an understanding of how the systematic relationship between situations and language can be found, represented, and used to tailor generated texts.

3 Register theory

As the theoretical background for this research, we use the constructs of *register theory* [19] as developed within Systemic-Functional Linguistics [18]. This body of theory is concerned precisely with the interrelationships between linguistic variation and types of audience and situations. According to register theory, not only do different situations and users affect language, they do so in systematic and specifiable ways. The different kinds of language called for in varying situations are called *registers*. Thus, registers may be seen as describing the 'argots' used by different classes of users. Based on extensive studies of language [8, 12, 13, 20, 21, 22, 34, 36, 60, 61, 64], the general aim of the theory is to make *explicit* the links between situational features and linguistic features and to specify the linguistic consequences of using language in particular situations.

It is important at this point to note that, throughout the rest of this paper, we will use the term *situation* in a very broad sense. In that sense, it includes characteristics of both hearers and speakers as these contribute to a detailed classification of the 'type' of situation that prevails at any time. We see this term as a generalization across previous notions of speaker, hearer, and user models.

Register theory, with its theoretical inclusion of *all* situational factors which can *systematically* influence linguistic variation, e.g., communicative setting, type of audience, speaker-hearer social relationship, etc., makes immediate contact with many issues in user modeling. As a technical construct, register is often claimed to decompose informal categories of sources of language variation, such as, for example, 'expert/novice', 'formal/informal', or 'written/spoken' that are now also found in computational models, into sets of distinctions whose consequences for linguistic expression are formally represented and whose motivations in terms of the communicative situation can be clearly stated. Register theory suggests the possibility of building into computational models explicit recognition of and sensitivity to the varieties of language use that are distinguished by a language community. This is very desirable.

Registers usually defined in terms of selections of situational features that call for the selection of particular corresponding linguistic features. These situational features organized into a network of interdependent choices. Each choice may constrain the linguistic alternatives available in the discourse organization and the grammar. (This will be illustrated in Section 5.1.)

Linguists have worked out registers for specific situations and have determined their linguistic consequences. Although the theory is not worked out in sufficient detail to immediately provide a computational implementation for it, these results can be used to guide further investigations. In particular, using register theory as a theoretical background for our research gives us several advantages:

- **A terminology for representing dimensions of situational variation.** Through extensive linguistic analyses of texts for particular situations, linguists have already determined some important dimensions of situational variation that affect language. These are often rather fine distinctions that can only be established on the basis of extensive text studies. Some examples of such features found in situations concerned with getting people to do things and making offers are: *suggestive vs nonsuggestive*, *pointed vs muted*, *assertive vs consultative* [21]. These features are important for user modeling research as they indicate the types of information that need to be modeled in order to tailor language. Furthermore, besides providing us with a vocabulary, these analyses also give us guidance in the type of information that we should be looking for in analyzing situations.

- **Some identified types of linguistic variation that occur as consequences of variations in situations.** Situational features can place constraints on the grammatical forms that occur. Two types that we have found in our own work are briefly described in Section 4. Other examples, taken again from Hasan's analysis of situations of making offers and 'demanding goods and services' [21], include: the selection of particular speech act types; particular lexical selections (e.g., *want vs like*); complexity of verbal expressions (e.g., *do you want ... vs would you like ...*); selection of negative or positive polarity; and many others. These types of linguistic variations can guide us in our linguistic analysis. They also provide the basic set of kinds of constraints that the mechanisms we implement for grammar control must support. It is noteworthy that this fine level of phrasing motivation would not be possible without an extensive model of grammar and grammar control. It is also important to note that a register network not only contains the network of situational features but also has associated with each choice its linguistic consequence.

- **A well-established methodology for seeking patterns of variation that co-vary with situation**. The theory aims at drawing the *finest possible distinctions* among situations that have some effect on language. These distinctions are uncovered by detailed linguistic analyses of the types of language that occur in various situations. When systematic differences between two types of language are found, one can posit a controlling situational feature and search for re-occurrences of the set of linguistic features it controls within other situations. It is important to try to uncover minimal distinctions. This is the means by which rather general situations can be broken down into component situation-types that have determinate consequences on language.

- **A framework in which the results can be cast**. According to the theory, the dimensions of situational variation are represented as systems of choice, whose interdependencies are captured by linking the systems into networks (the *register networks*). The results of making particular situational 'settings' or choices in the register network then have the effect of preselecting (or of limiting) alternatives of expression in the grammar network. This provides a possible representation for user characteristics and for their interaction with a computational system exploiting them. This gives rise to the architecture shown in Figure 5.[2]

We are adopting the methodology and framework proposed in register theory in order to uncover the similarities among situations and the links between situational features and linguistic features. We are also taking existing linguistic work to suggest the kinds of classifications of situations that are needed and their likely influence on language. By using a network of register — or situational — features to constrain language, it ought to be possible to systematize the statement of effects of particular situations on language and thus achieve tailoring. When applied to user modeling and natural language generation, then, register theory suggests a framework that:

- provides more structure to the statement of the dimensions of user models (that affect language) than the independent dimensions of variation that have been prevalent formerly;

- clarifies the types of constraints that situations can impose on linguistic expression.

[2]This architecture has already been adopted by Patten in [50].

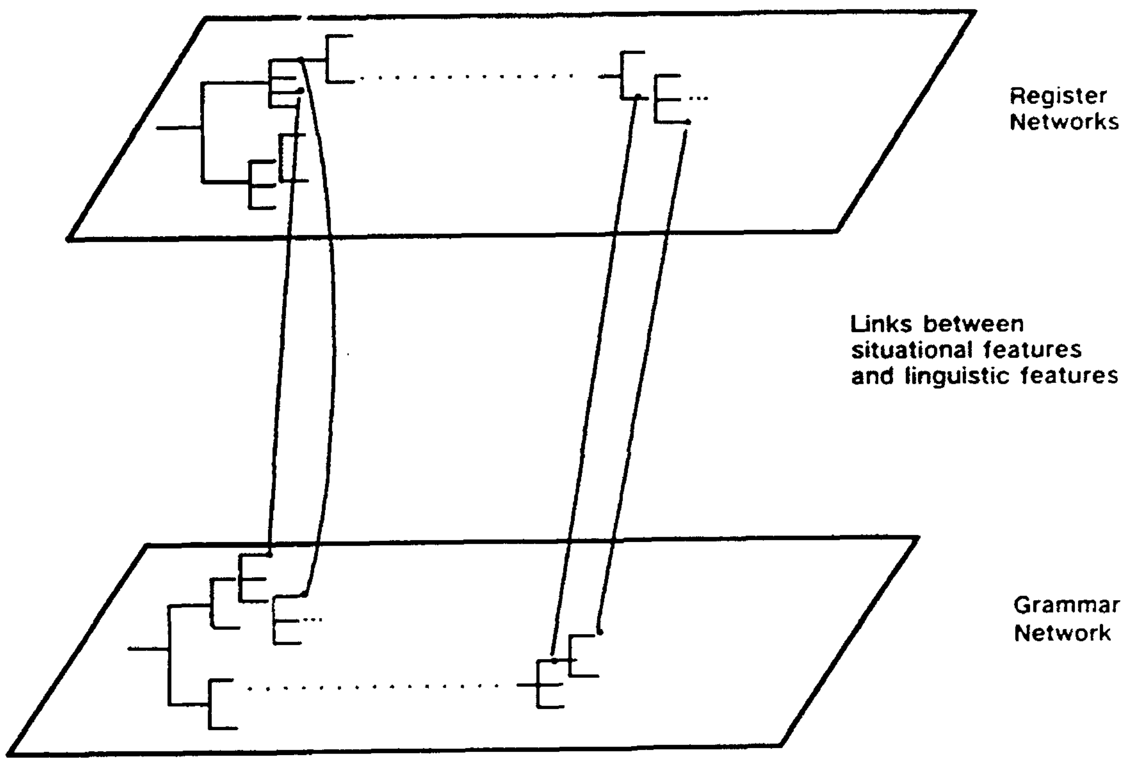

Figure 5: Architecture for controlling phrasing

4 Our Pilot Study

In a pilot study, we gave a text planning system a limited ability to control fine-level grammatical details in response to a specific situation and were able to provide a specification of the type of language required in a given situation. Our approach was to restrict our attention to the particular kinds of language required in a single domain and construct a set of particular situational features that were used to control the grammar. In this section, we briefly describe the computational context within which we generated situationally controlled language and the distinct mechanisms that we found necessary for enforcing that control.

4.1 The Explainable Expert Systems Project: EES

The system for which we are generating language is an expert system constructed using the Explainable Expert Systems (EES) framework [45, 59]. The goal of the EES project is to automatically produce expert systems that are capable of explaining their reasoning as well as the content of their knowledge base *in appropriate natural language*. In this framework, an expert system includes an extensive quantity of *support knowledge* for explanations, including, for example, terminology definitions so that the system can explain specific terms it employs. The system we used in our preliminary experiments is an expert system designed to diagnose digital circuits. Plans are also under development to work with expert systems in the domain of network communications.

The expert systems being developed are expected to support interactions with different types of users. In our pilot study, we considered three user types: system developers, end-users, and students using a system as a tutoring aid. In our current work, we are refining our classification of these groups and adding other groups as the systems are used. The three initial groups can be described in the following terms:

System developers: These users want to make sure that the knowledge base is correctly represented and that the system is working properly.

End-users: These users want to follow the system's reasoning but do not know much about expert system technology (or even about computer science).

Students: These users are naive users trying to get acquainted with an application domain.

```
(define-type-attributes faulty-system
    :defining-conditions: ((and (there-exists (o in (output-part self))
                                      (not (equal (expected-value (signal-part o))
                                                  (actual-value (signal-part o)))))
                           (for-all (i in (input-part self))
                                    (equal (expected-value (signal-part i))
                                           (actual-value (signal-part i)))))))
```

Figure 6: Defining conditions of a `faulty-system`

4.2 Linguistic variation required

To perform our studies, we asked the potential users of the system to provide the paraphrases of the concepts in the knowledge base and of problem solving behaviors that they would like the system to give them. From this corpus, we studied the different forms of language that would be required for the expert system to interact with these three different user types. We found two distinct kinds of linguistic variation, which are primarily differentiated by the distinct mechanisms of control they require over the grammar: *experiential* variation, which requires the definition of a sub-grammar, and *head-modifier* variation (or *logical* variation), which requires a mechanism that constructs a head-modifier structure given a specification of allowable heads. We implemented these mechanisms in our system. Some examples of texts generated from the same underlying representation (see Figure 6) using these mechanisms are shown in Figure 7. These texts are all appropriate responses to the question: 'What is a faulty system?' in the digital circuit diagnosis context, but they are differentiated according to the type of user involved. A further set of contrastive examples also generated by our system is given in Figure 8.

We now discuss briefly the two types of variation we found and the mechanisms we implemented to support them.

Ideational variation. In examples of the language used for interactions with system developers (the (a)-texts in the figures), our preliminary analysis indicated some specific language variation such as the following:

- Processes of existence were lexicalized by the lexical item *exist*, rather than some more neutral item (such as *be*, as in *there is a book*).
- 'De-emphasizing' expressions, such as the possessive modification *X's Y*, were avoided in favor of the more equal emphasis of *Y of X*.

- Pronominalization was not attempted.

This style contrasts with the language used with end-users (shown in the (b)-texts) where, for example, pronominalization is readily allowed, and possessives such as *X's Y* appear. All of these aspects of the phrasing may be controlled by choosing appropriate grammatical features among the possible features available in the grammar.

Logical Variation. The other type of linguistic variation is concerned primarily with the dependency relationship that holds between heads of phrases and their modifiers. The selection of precisely which elements are to serve as heads of phrases, and which have to serve as modifiers, has a decisive influence on the meaning that may be expressed. Consider for example the texts shown in Figure 7. When we examine texts (5a), (5b), and (5c), and the input specification that gives rise to these sentences (Figure 6), we can see that one of the most striking differences between them is the way that the components of the propositional content are allocated to head and modifier status. For example, from the input specification, we can see that the quantifiers of the input are expressed in very prominent positions (as process, 'there exists', for example) in the predicate calculus-like language of text (5a). However, they appear much less significantly in texts (5b) and (5c). They do still appear, however, as determiners (*one, all*), or in specifying whether a noun should be singular or plural.

The selection of heads is neither random nor unsystematic. Particular situations highlight certain entities and relations at the expense of others. This must be respected if natural phrasing for that situation is to result. Given a statement of these entities, their relations, and their relative rankings, we can control the phrasing phase of the generation process in terms of their allocation to appropriately ranking heads. From the particular kinds of language required in our domain, we constructed a set of 'terms' (or available heads), together with their rankings, to be employed when generating language for particular users. We then designed an algorithm that controls this type of allocation based on the audience of the text. Given that this phase of the phrasing problem is one of selecting at a *fine* level of detail which information to express to the user, this can be seen as text planning at a very fine-grain level.

4.3 The initial implementations

Two forms of mechanism support the kind of linguistic variation that we have just described and that are illustrated in Figures 7 and 8:

```
The system is faulty, if there exists a O in the set of the output
terminals of the system such that the expected value of the signal part of
O does not equal the actual value of the signal part of O and for all I in
the set of the input terminals of the system, the expected value of the
signal part of I equals the actual value of the signal part of I.
```

Text generated for system developers (5a)

```
The system is faulty, if all of the expected values of its input terminals
equal their actual values and the expected value of one of its output
terminals does not equal its actual value.
```

Text generated for end-users (5b)

```
The system is faulty, if its inputs are fine and its output is wrong.
```

Text generated for students (5c)

Figure 7: Examples of variations generated from the same propositional input for different user types

```
For all I in the set of the input terminals of the system, there exists a
CI in the set of the input terminals such that there exists a C in the set
of the components of the system such that the C has a input terminal such
that the CI is connected to the I.
```

Text generated for system developers (6a)

```
All of the system's input terminals are connected to an input terminal of
one of its components.
```

Text generated for end-users (6b)

```
The input terminals of the system are connected to its components' input
terminals.
```

Text generated for students (6c)

Figure 8: Examples of text generated for different user types

- the ability to automatically constrain the grammar to a *sub-grammar* that only generates language exhibiting the features required for a given register. A sub-grammar is constructed by allowing only the selection of the grammatical features required for a given register;
- a definition and ranking of *register terms* (i.e., heads available in a particular register) that define preferences for head-modifier dependency relationships.

With these two mechanisms, our text planner is now able to systematically control the phrasing of a given propositional content by choosing a register. Thus, since the register specification features include details concerning types of speaker-hearer relationships, tailoring on the basis of user characteristics is made possible. See [2] for further details on the implementation and mechanisms.

5 Work in progress: methodology proposed

While the mechanisms we have developed give some of the behavior required for tailoring the phrasing of a text, our set of situational features was very restricted. We are currently building on our preliminary study to create a framework in which it will be possible to express the knowledge needed to support language use in any context and systematically gain control over phrasing. We are extending our set of situational features to develop a language that can be used to specify the general characteristics of situations that have an influence on linguistic realization. We are also developing further the mechanisms necessary to gain control over fine-level grammatical details. We will then test our framework by creating registers for a few particular situations in the context of the expert system explanation facility. We now propose a methodology based on register theory for further developing an understanding of how situations affect language, i.e., for identifying the dimensions of variations of situations and representing them in a way that facilitates tailoring.

5.1 How to capture situational generalizations: building register networks

The mechanisms described for the pilot study did not yet capture generalizations across more or less similar situations. It is this aspect of register theory that is crucial. According to the theory, registers are defined in terms of selections of situational features that call for the selection of particular corresponding linguistic features. More formally, the situational features that define registers are organized into a *network*

of interdependent choices, each of which may constrain the linguistic alternatives available in the grammar by requiring specified sets of linguistic features to occur (cf. Figure 5). Each choice in the network is designed to draw the finest possible distinction between two contrasting forms of expression possible at that point. These choice points form the generalizations that the register theory is seeking, and the basis for building a register network. A register network thus captures the dimensions along which situations differ and resemble each other (i.e., generalizations across situations) explicitly in terms of their consequences on language.

The principle of minimum distinctions is important, as it is the means by which rather general situations are broken down into sharable components of situation-types that have determinate consequences on language. For example, it would be unlikely for the linguistic consequences of a situation-type such as **relaxing over coffee with a co-worker** to be statable simply in terms of constraints of the kind we have presented here. The language used will vary quite considerably depending, for example, on the status of the co-worker: relaxing with a superior, a person of equal status, or with a subordinate will usually be associated with different registers, possibly sharing some situational features (as well as sharing some situational features with yet other registers). Thus, this situation-type is too vague and must be decomposed along dimensions whose linguistic consequences *can* be stated: for example, dimensions such as the relative social status of the interlocutors, the importance of the communicative situation, the physical location of the interaction, means of communication (face-to-face, telephone, etc.), topic area of the communication, etc. Only in terms of these more basic classificatory features are regularities in the relationships between situation and language likely to be found. Actual situations will then typically contain many component situation-types: i.e., they will be fixed at many points in a multi-dimensional space of situation possibilities.

The classificatory features that are needed to build a register network must come from detailed linguistic analyses of the types of language that occur in various situations; that is, the construction of the register network must be the result of empirical investigation. Stating *ad hoc* generalizations concerning phrasing without sufficient linguistic evidence is unlikely to be successful. When analyzing the data, we can posit a controlling situational feature when systematic differences between two types of language are found. We can then search for re-occurrences of the set of linguistic features that situational feature controls within other situations.

Our current three-way distinction of user types is therefore inadequate as it does not factor all the linguistic consequences of the distinct communicative situations and fails to capture many relevant generalizations and points of contrast. In our pilot study, we have however started to uncover some generalizations. For example, the language of the system developers and that of the end-users have more in common

(i.e., there are more constraints on the grammar that occur for both user types) than do either with the language for students. We are now factoring out these commonalities in order to state them once in a network, in the form of a single situational feature that is shared by both the system developer and the casual user. As more data is analyzed, we expect the depth of detail of these generalizations to grow considerably. We are now trying to identify in more specific terms than we used in our pilot study the range of language required by users of the expert systems built in these domains, gathering corresponding text both from written sources and from sessions with the users whenever possible. Drawing on existing work in register theory that suggests dimensions that are relevant, we then perform the necessary linguistic analysis to uncover both situational and linguistic variations by applying the minimal distinction methodology suggested by register theory. The resulting network allows us to factor out commonalities and to state dimensions of differentiation more effectively, in terms of linguistic consequences of each situational feature.

An example of a register network is shown in Figure 9.[3] This network is taken from Hasan's analysis of situations of 'making offers and demanding goods and services' [21]. It shows the situational features that need to be differentiated as they have an influence on language (e.g., *suggestive vs nonsuggestive*, *pointed vs muted*, *assertive vs consultative*). From this network, we can see clearly that situational features are organized into a network of interdependent choices. Importantly, the network also *indicates the constraint on* the grammatical forms that each choice has on language. These constraints, such as the selection of particular speech act types or of particular lexical items, are indicated by numbers and letters on the links in the Figure. With such a network, one clearly sees both which aspects (features) of the situation have an effect on language and the effect each feature has on language. If each possible effect on language is implemented to constrain the linguistic resources, then control of phrasing can be obtained in a systematic way by specifying the features of the situation at hand.

More specifically, then, we are proposing the following methodology for studying the tailoring that needs to take place when generating language. This methodology is derived from the notional schema for investigating registers that has been followed for years [3].

1. Linguistic analysis of sample texts from selected application domains to discover the required text and grammatical organizations. We have already mentioned the importance of linguistic analysis to reveal linguistic variation. Importantly,

[3] *In all of the network figures the following conventions are used:* '|' indicate that the paths are disjoint; '{' indicates that the consequences from both paths are to be followed; and '}' indicates that all the selections to the left are preconditions for the selection to the right.

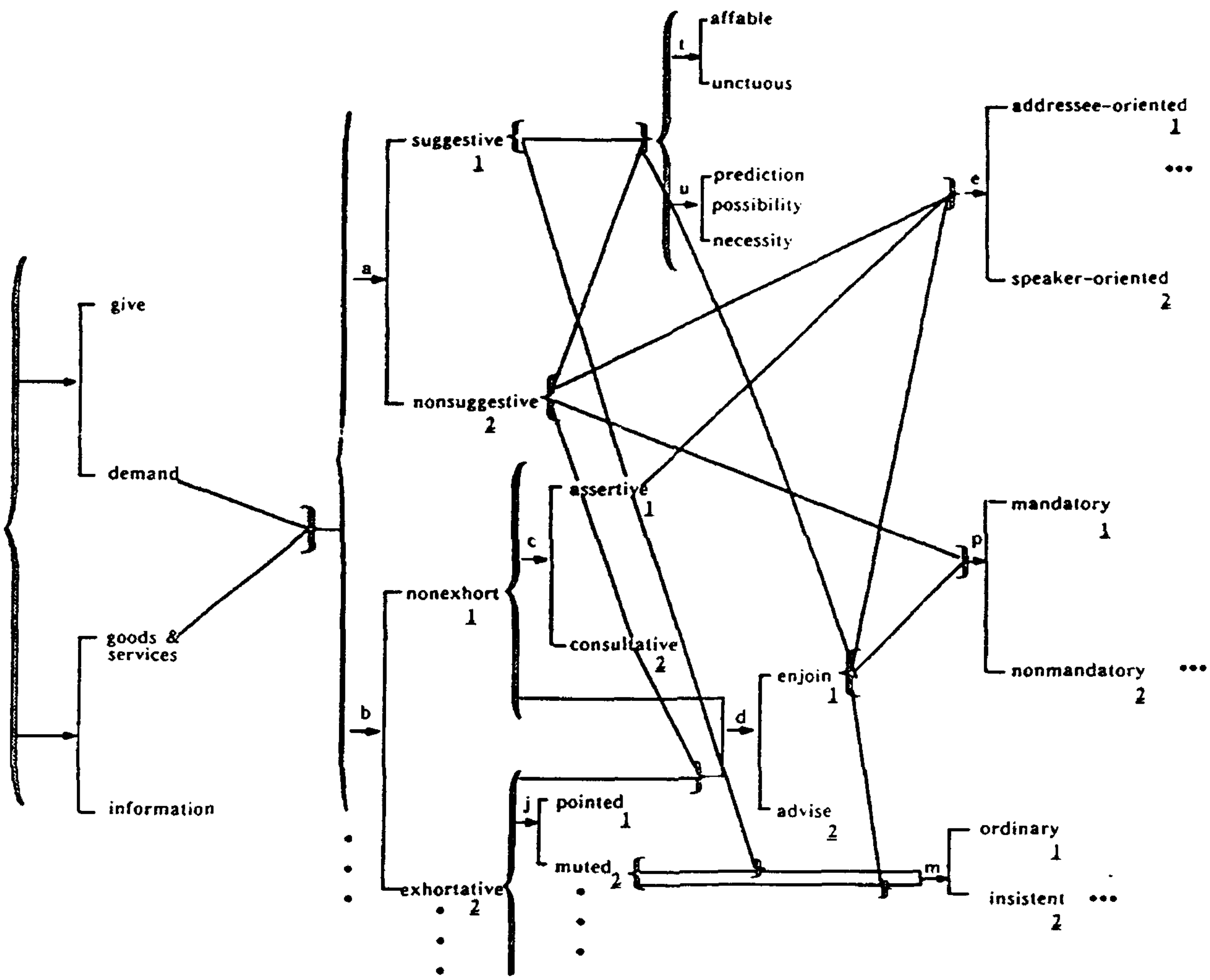

Figure 9: A portion of a register network: semantic options in making people do things [21]

this analysis must be carried out in terms of an *explicit* grammar (we are using the systemic-functional grammar NIGEL).[4] This provides a solid base for the statement and testing of predictions of situational effects on linguistic realization.

2. Constructing a factorization of the properties of the text and clause organizations — as revealed by the analysis of the previous step — that groups them along dimensions of similarity and dissimilarity. For the grammar, this is a natural consequence of performing the analysis with respect to a systemic-functionally based grammar such as NIGEL, since this is precisely the mode of organization employed within the definition of the grammar.

3. Constructing a correlation between properties of texts and clauses and aspects of the users and situations for which those properties occur. In general, our approach is to restrict our attention to the particular range of language variation required for the responses that an expert system is to give for specified types of users. We then work *back* from these types of language in order to construct a *specification* of the important classificatory features of the situation. These then constitute the register definition, which describe the situation. They are used as an additional source of input to the text generation process to provide effective control of phrasing whenever language is to be generated in a situation of the recognised type.[5] For preliminary data sets of limited size, approximate correlations of this kind performed manually are adequate; for later stages of the research where large data sets are to be considered, statistical clustering analyses may be necessary to exhaustively identify possible candidates for controllable systematic variation.

4. Expressing the applicability of those constraints in terms of a classificatory register network of communicative situations.

5. Formal specification of the mechanisms of interaction between register definitions and the grammar and text organization.

6. Testing, evaluating, and refining the definitions of register constructed and their implementation in the face of their performance.

[4]At this point in the research, we are concerned mostly with variations in phrasing. We will later look towards applying this methodology to text organization as well. There, we will need to use accounts of text organization as explicit as possible, e.g., Mann and Thompson's *Rhetorical Structure Theory*, Hasan's *Generic Structure Potential*, and Martin's *Conjunctive Relations*.

[5]In future work, these will also act as constraints on for the selection of rhetorical strategies. See Footnote 5.

While in this research, we concentrate more on the relation between possible register specifications and the control of *grammatical* resources, we will also attempt in the future to allow for significant interaction between the register specifications and the permissible text organizations. Here we will be incorporating previous computational work on the relation between contexts of use and text organization (e.g., Paris 1987, 1988). At present, reflecting the state of many generation systems, the generation process is broken down into two steps: the first selects some area of content and provides some textual, large-scale organization for the expression of that content; the second concentrates on selecting at a fine level of detail exactly which aspects of the propositional content to express or highlight, and selecting appropriate lexicogrammatical expressions of that content. Clearly, both of these steps needs to be sensitive to register issues, as illustrated in Figure 10, although in this work we are addressing primarily the second phase.

5.2 Building our register network and specifying the constraints we require

We move from our initial three way distinction to a network-oriented description as follows. Based on our analysis, we first attempt to set up situational features which appropriately classify the situation in which language is being generated. Note that in the early stages of investigation, we will only obtain 'sketch' networks, in which 'situational features' are not to be considered final or, in some cases, even appropriate: they stand as placeholders during the successive refinements that the investigation of more data will require. In a sense they can be viewed as shorthand notation for *sets* of actual situational features that are still to be uncovered. We then specify the constraints that these features bring to bear on the language generated in the corresponding situation. This allows us to factor out commonalities and to state dimensions of differentiation more effectively.

Based on our initial analysis, one type of constraint that a situational feature can add is the *inclusion* of a set of *register terms*. For example, in the domain of the digital circuit diagnosis, domain concepts (such as *signal-part* and *connected-to*) and a set of terms defined in predicate calculus (such as *not*, and *forall*), are available for constructing language. This could be represented in a register network in a way analogous to the insertion realization statements in the NIGEL grammar, as shown in Figure 11.

Note that two terms at least can be moved up further upstream in the network: *exist* and *forall*. These terms correspond to any formal type of field, not just the digital circuit diagnosis domain. Moving these terms upstream in the register network

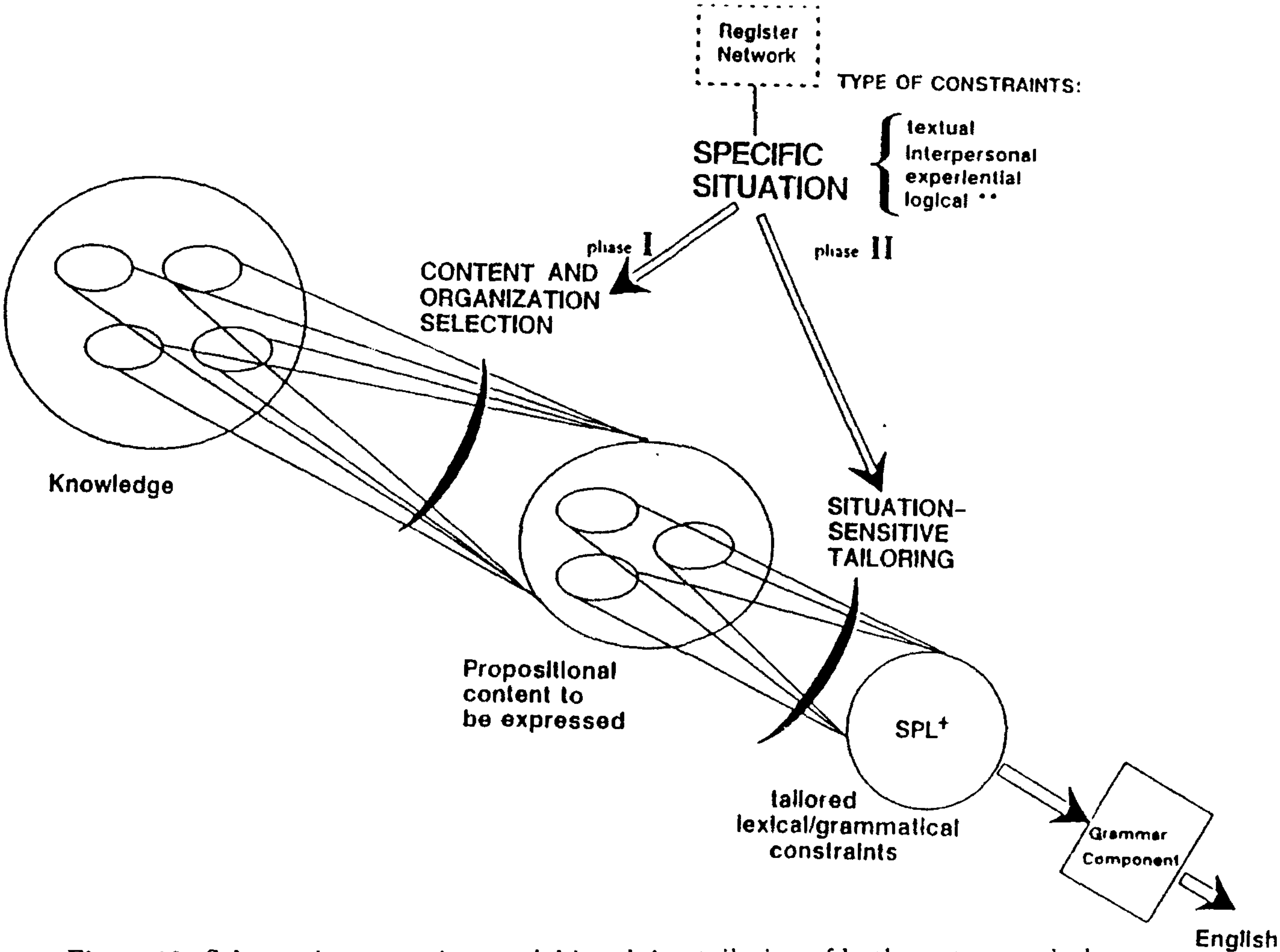

Figure 10: Schematic generation model involving tailoring of both content and phrasing

```
    |--> DCD-type-field-of-discourse ...
    |    [ + (exist forall not equal
    |         object-relation-ascription
    |         material-property-ascription
--->|         signal-part connected-to
    |         expected-value actual-value value signal
    |         output-terminal input-terminal system variables
    |         input-part output-part part-of) ]
    |
    |--> non-DCD-type-field-of-discourse ...
```

Figure 11: Planned definition of register terms - inclusion

```
---> DCD-type-field ------------}
                                }---> DCD-predicate-calculus
---> predicate-calculus-like ---}      [ (ifthen iff and or) >
                                         (exist forall not equal
                                          object-relation-ascription
                                          material-property-ascription
                                          signal-part connected-to) >
                                         (expected-value
                                          actual-value value signal
                                          output-terminal input-terminal
                                          system variables) >
                                         (input-part output-part
                                          part-of) ]
```

Figure 12: Planned definition of register terms - logical constraints

would avoid multiple statements and mark the similarities among all formal situations. Only terms that are differentiated from the terms included by other alternatives at any point in the register network should be located at that point.

The other type of information used in our previous definition was the *ranking* information. This we factor out and specify separately, although again as constraints licensed by particular selections of situational features from the register network. An example specification is shown in Figure 12. Here we use another realization operator, denoted by ">", that indicates relative *logical* ordering of head selection. In the figure, for example, the predicate calculus terms **exist, forall, not** are given a higher rank than terms such as **expected-value** when generating language very close to predicate calculus in the digital circuit diagnosis domain. Again, this does not make any use of possible redundancies that could be factored out and expressed at a less delicate level in the register network.[6] We are now working towards this factorization.

From our initial implementations, we have already identified two more mechanisms that need to be implemented.[7] It is likely that we will need further types of realization statements for register networks. This will be informed both by our own analysis of data and by ongoing work in register theory. We are working towards a building a network which defines a set of interdependencies between features, each of which may have associated with it a set of realization statements which impose constraints on the grammar's generation. The realization constraints are thus selectively employed when traversing a register network in order to build up a set of constraints that are to apply when generating text for the types of situation specified by that traversal. By constructing such a network of registers to constrain language, we thus hope to systematize the statement of effects on language of particular situation-types.

6 Tailoring in Computational Linguistics

Two main areas of research have addressed problems similar to ours: (1) user modeling, where researchers have been concerned with building (e.g., [6, 5, 7, 29, 30, 32, 33, 53]) and exploiting (e.g. [1, 4, 28, 37, 38, 47, 39, 62, 65]) user models to aid systems in making various decisions required in the course of interacting linguistically; and (2) pragmatic goals in generation, where researchers have developed computational accounts of language variation depending on pragmatic information such as *speaker-hearer relationships* [25, 55]. The work described in this paper furthers the work

[6]In fact, the ordering mechanism as we describe it does not allow effective factoring out of ordering information at present. We are now working on a refined version.

[7]The interested reader is referred to [3] for further details on these mechanisms.

on tailoring in two major ways: (1) It addresses a problem that has not really been studied before, that of tailoring the phrasing of a text. (2) It proposes a methodology to study more rigorously the various aspects of users and situations that need to be modeled (for language use) and their effect on language. We present each related area in turn, and discuss how our work differs from but builds on these previous approaches.

6.1 User Modeling

Research in user modeling has to date focused mainly on the problems of representing and inferring the user's goals, plans, wants and beliefs (e.g., [4, 44, 5, 39, 30, 32, 33]), recognizing misconceptions (e.g., [28, 37]), and superposing various stereotypes (e.g., [53, 7]). There has also been some work on trying to characterize how language is affected by various user models (e.g., [40, 47, 65, 62]). For the most part, this body of research has been concerned with choosing the appropriate content from the knowledge base and organizing it, based on a user's level of expertise or his/her goal for using the system. After a text plan is constructed with the help of a user model,[8] a 'dictionary interface' typically assigns lexical items and syntactic structures to the various propositions, and a generator is then responsible for actually producing English sentences (e.g., [38, 47, 43]). The phrasing problem has not been addressed in a general way.

6.2 Pragmatic goals; grammar control

One of the most impressive generation programs capable of varying phrasing is Hovy's PAULINE [24, 25], which offers a striking computational demonstration of the importance of providing for sensitivity to pragmatic information in generation.

Hovy organizes planning/generation around a set of twelve *rhetorical goals* that critically influence both content selection and phrasing. Based on the values assigned to the various rhetorical goals, PAULINE can generate greatly differing text about the same event. Rhetorical goals are appealed to during generation in order to decide among possible alternatives in grammatical phrasing that are not distinguishable propositionally. The grammar explicitly consults the values of rhetorical goals, such as *formality* or *fluidity*, when a choice is to be made. This mechanism requires that the grammar include explicit statements of the stylistic importance of its alternatives.

[8] At this stage, a text plan consists of an organized collection of propositions the planner has decided to express.

The rhetorical goals defined in PAULINE serve the same function as register features in the approach we are proposing. However, while rhetorical goals can be seen as a way of characterizing situations, they provide relatively gross descriptive categories for the language occurring in those situations. This complicates the treatment of a rhetorical goal's influence on the language that is generated for the reasons we described in Section 5.1: it is necessary to find minimal alternations so that their linguistic consequences can be stated with maximal lack of redundancy and precision. Since a single goal may affect many issues in the grammar, there is the danger in PAULINE that, when new situations are considered, they will not group together the required linguistic constraints in a way that is compatible with groupings previously defined. This would then require redefinition of the rhetorical goals affected. Although it would be possible to pursue this development, we feel that the explicit guidance offered by a prior commitment to finding minimal register alternations, and representing the distinctions that they draw in network form, improves on this methodology. It also allows us to state very clearly just what the available forms of constraints between situational features and linguistic realization are, and to maintain the independence of the grammar.

6.3 Limitations we are trying to address with this research

There are two main limitations we are trying to address in this work, using the methodology proposed by register theory to guide our investigations. First, current results on pragmatic goals or on tailoring have provided little in terms of *systematic* ways of capturing the *linguistic consequences* of user models. Indeed, the largely independent dimensions of variation found in existing computational models of generation have not yet provided any general framework for understanding the phenomena of the interaction between language and context. They have not gone particularly far in revealing possible *systematicities* between differing *types* of dimensions of user/pragmatic variation and differing *types* of language variation. Register theory, as developed in the systemic-functional linguistic tradition, maintains that there is, in fact, a rich area of systematicity which deserves description and explanation. If this is so and this aspect of language could be brought centrally into computational models of generation, then both theory construction and system functionality would benefit. Furthermore, research in user modeling and 'pragmatic' goals, has typically been restricted in that the dimensions along which users and situations can be classified have been studied in isolation as largely independent. We expect the explicitly accepted likelihood of extensive interactions between situational features to improve the research methodology.

Second, the phrasing problem has not been addressed in a general way, and control

of the phrasing task has mostly remained below the level of detail that the text planning process has under theoretical control. As we mentioned above, after a text plan is constructed with the help of a user model, a 'dictionary interface' typically assigns lexical items and syntactic structures to the various propositions. The input to the dictionary interface is rarely detailed enough to completely control the many possibilities for expression that current grammar components provides, however; there is, in fact, a large gap between the level of detail of the output of the text planner and what is necessary to fully control a grammar.

A number of researchers have already recognized the need to gain control of the fine level of grammatical detail required for varying the phrasing of a message (e.g., [25, 41, 27, 56, 55, 54]). However, only very specific aspects of the phrasing have been addressed, such as planning an appropriate referring expression or choosing an appropriate cue phrase (e.g., [15, 57, 51, 1, 16, 23, 1, 9, 46, 52, 10]), and, often, how the phrasing is controlled depending on the situation has often not been made explicit. Indeed, little work has been done in specifying ways for a text planner to *systematically* control the various alternative phrasings available for any particular propositional content in response to the particular type of audience and communicative situation. Furthermore, necessary separation between text planning and realization has not been maintained. This separation is crucial as a text planner should not maintain detailed knowledge of the grammatical possibilities offered by the grammar [26, 48]; to do so complicates the planning process considerably by requiring the text planner to concern itself with details from an inappropriately low level of abstraction. Similarly, the grammar should not include detail at an inappropriately 'high' level of abstraction.

In this work, we aim to provide a general phrasing control component that interfaces between the output of the first phase of planning and the input to the grammar. This component is to decide both *which aspects* of the output of the first phase are most appropriate for each user type and *how they are to be phrased* for that user. (This component can be seen as furthering the text planning process.) It is only the result of this second phase of selection/organization that provides sufficient guidance to the grammar and lexical selection components. This was already illustrated in Figure 10.

It is important to point out, however, that we will build upon all the previous work described in this section and attempt to integrate them into a large generation system like Penman in a coherent, unified framework, using the rather general mechanisms of registerial control as an organizational framework for specifying linguistic consequences of situational classifications. In this research, we are attempting to combine the strength of computational models (the degree of explicitness they require) with that of the linguistic approaches (a rigorous methodology) in order to overcome their weaknesses [3].

7 Conclusions

In this paper, we first presented a problem that has not been addressed in computation models of generation: tailoring the phrasing of a text depending on the intended user and the situation. Indeed, while there has been work on trying to characterize how language is affected by various user models (e.g., [37, 40, 47]), for the most part, this body of research has been concerned with choosing the appropriate content from the knowledge base and organizing it. After a text plan is constructed, a 'dictionary interface' typically assigns lexical items and syntactic structures to the various propositions, and a generator is then responsible for actually producing English sentences (e.g., [38, 43, 47]). The phrasing problem has not been addressed in a general way. This is the problem we are addressing with the research presented here.

We showed how relevant linguistic studies can be brought to bear on the problem of user modeling and tailoring. Much literature in systemic-functional linguistics has been devoted to precisely the issue of interaction of situations (and users) to language. We might then expect that there could be a significant contribution from systemic linguistics, and, in particular, from register theory in this area. In this work, we investigate this suggestion in the light of an existing computational system with specific communicative needs.

Based on a specific linguistic theory, we proposed a methodology to systematically study the problem of tailoring phrasing: we feel that the explicit guidance offered by a prior commitment to finding minimal register alternations, and representing the distinctions that they draw in network form, provides a good methodology for building and exploiting user models. It also allows us to state very clearly just what situational features are important in terms of language and how they affect language. A further improvement in research methodology is offered by the explicitly accepted likelihood of extensive interactions between situational features. While research in user modeling and 'pragmatic' goals, has typically been restricted in that the dimensions along which users and situations can be classified have been studied in isolation as largely independent, we expect that register networks will be complex.

We believe that register theory can benefit user modeling research, by proposing features of user models that will be need to be captured: indeed, uncovering situational features that affect language is equivalent to uncovering features that a user model needs to contain in order for appropriate tailoring to occur.[9] Furthermore, the theory proposes a representation for some features of a user model and their possi-

[9] It is also interesting to point out that register theory will also gain from this attempt at employing it in a computational setting: indeed, the computational setting will enforce a level of explicitness not required previously [3].

ble interactions with the rest of a computational system. We thus believe that the interaction of these two research areas will be a very promising one.

Acknowledgements

This paper and the work underlying it has been improved considerably by the detailed criticisms of Bill Swartout, Johanna Moore, Eduard Hovy, and Erich Steiner. We also thank Michael O'Donnell for his comments on earlier versions of this paper and the anonymous reviewers of the Workshop on User Modeling to which an abstract of this paper was initially submitted for their useful comments. Finally, we appreciate the helpful discussions of the participants of the Systemic Congress which helped us developed our work. The mistakes, as always, remain ours. The research described in this paper is supported in part by the Defense Advanced Research Projects Agency (DARPA) under a NASA Ames cooperative agreement number NCC 2-520, by the AFOSR contract F49620-87-C-0005, by the DARPA contract MDA903-87-C-641, and by the NSF co· tract IRI-9003087. The opinions in this report are solely those of the authors.

References

[1] Douglas E. Appelt. *Planning Natural Language Utterances*. Cambridge University Press, Cambridge, England, 1985.

[2] John A. Bateman and Cécile L. Paris. Phrasing a text in terms the user can understand. In *Proceedings of the Eleventh International Joint Conference on Artificial Intelligence*, Detroit, Michigan, 1989.

[3] John A. Bateman and Cécile L. Paris. Constraining the deployment of lexicogrammatical resources during text generation: towards a computational instantiation of register theory, 1990. Presented at the Sixteenth International Systemic Congress, Helsinki, Finland; to be included in selected papers from the conference.

[4] Sandra Carberry. Tracking user goals in an information-seeking environment. In *Proceedings of AAAI-83*. American Association for Artificial Intelligence, 1983.

[5] Sandra Carberry. Plan Recognition and its Use in Understanding dialogue. In Alfred Kobsa and Wolfgang Wahlster, editors, *User Models in Dialog Systems*. Springer Verlag, Symbolic Computation Series, Berlin Heidelberg New York Tokyo, 1989.

[6] David N. Chin. User Modelling in UC, the UNIX consultant. In *CHI'86 Proceedings*. Computer Human Interactions, 1986.

[7] David N. Chin. KNOME: Modeling What the User Knows in UC. In Alfred Kobsa and Wolfgang Wahlster, editors, *User Models in Dialog Systems*. Springer Verlag, Symbolic Computation Series, Berlin Heidelberg New York Tokyo, 1989.

[8] Carmel Cloran. Negotiating new contexts in conversation. *Occasional Papers in Systemic Linguistics*, 1:85–110, 1987.

[9] Robert Dale. The generation of subsequent referring expressions in structured discourses. In Michael Zock and Gérard Sabeh, editors, *Advances in Natural Language Generation: An interdisciplinary perspective; Volume 2*, pages 58–85. Pinter Publishers, London, 1988.

[10] Robert Dale. Cooking up referring expressions. In *Proceedings of the Twenty-Seventh Annual Meeting of the Association for Computational Linguistics*, Vancouver, British Columbia, June 1989.

[11] M. Gregory and S. Carrol. *Language and Situation*. Routledge and Kegan Paul., 1978.

[12] Michael Gregory. Aspects of varieties differentiation. *Journal of Linguistics*, 3:177–198, 1967.

[13] Michael Gregory and Karen Malcolm. Generic situation and discourse phase, 1981. Mimeo. Applied Linguistics Research Working Group, University of York, Toronto.

[14] Susan J. Grimm. *How to Write Computer Manuals for Users*. LifeTime Learning Publications, Belmont, California, 1982.

[15] Barbara J. Grosz. The representation and use of focus in a system for understanding dialogs. In *Proceedings of IJCAI-77*, Pittsburgh, Pennsylvania, 1977. International Joint Conferences on Artificial Intelligence.

[16] Barbara J. Grosz and Candace L. Sidner. Attention, intentions and the structure of discourse. *Computational Linguistics Journal*, 12 (3), 1986.

[17] Robert Gunning. *The technique of clear writing*. McGraw-Hill, New York, 1968.

[18] Michael A. K. Halliday. *Explorations in the Functions of Language*. Edward Arnold, London, 1973.

[19] Michael A. K. Halliday. *Language as social semiotic*. Edward Arnold, London, 1978.

[20] Ruqaiya Hasan. Text in the Systemic-Functional model. In W.U. Dressler, editor, *Current Trends in Text Linguistics*, pages 228–246. de Gruyter, Berlin and New York, 1978.

[21] Ruqaiya Hasan. Offers in the making: a Systemic Functional Approach. Technical report; School of English and Linguistics; Macquarie University, 1984.

[22] Ruqaiya Hasan. The representation of meaning in the Systemic Functional model. Paper presented at the 16th. International Systemic Congress, Helsinki, Finland, June 1989.

[23] W. Hoeppner, K. Morik, and H. Marburger. Talking it over: The natural dialog system HAM-ANS. Technical Report ANS-26, Research Unit for Information Science and Artificial Intelligence, University of Hamburg, 1984.

[24] Eduard H. Hovy. Generating natural language under pragmatic constraints. *Pragmatics*, 11(6):689–719, 1987.

[25] Eduard H. Hovy. *Generating natural language under pragmatic constraints*. Lawrence Erlbaum, Hillsdale, New Jersey, 1988.

[26] Eduard H. Hovy. On the study of text planning and realization, August 1988. Presented at the AAAI-88 Workshop on Text Planning and Realization.

[27] Paul Jacobs. Why text planning isn't planning, August 1988. Presented at the AAAI-88 Workshop on Text Planning and Realization.

[28] S. Jerry Kaplan. Cooperative Responses from a Portable Natural Language Query System. *Artificial Intelligence*, 2(19), 1982.

[29] Robert Kass and Tim Finin. Rules for the implicit acquisition of knowledge about the user. In *Proceedings of the Sixth National Conference on Artificial Intelligence*, pages 295–300, Seattle, Washington, July 1987. The American Association of Artificial Intelligence.

[30] Robert Kass and Tim Finin. Modeling the user in natural language systems. *Computational Linguistics Journal*, 14:5–22, 1989. Special Issue on User Modeling.

[31] Martin Kay. Functional grammar. In *Proceedings of the 5th meeting of the Berkeley Linguistics Society*, pages 142 - 158. Berkeley Linguistics Society, 1979.

[32] Alfred Kobsa. A taxonomy of beliefs and goals for user models in dialog systems. In Alfred Kobsa and Wolfgang Wahlster, editors, *User Models in Dialog Systems*. Springer Verlag, Symbolic Computation Series, Berlin Heidelberg New York Tokyo, 1989.

[33] Alfred Kobsa. Modeling the user's conceptual knowledge in bgp-ms, a user modeling shell system. In *Proceedings of the Second International Workshop on User Modeling*. Sponsored by AAAI and the University of Hawaii, 1990.

[34] R. Mackay and A. Mountford. *English for Specific Purposes*. Longman, London, 1978.

[35] William Mann. An Overview of the Penman Text Generation System. Technical Report ISI/RR-83-114, USC/Information Sciences Institute, 1983.

[36] J. R. Martin and Joan Rothery. The ontogenesis of written genre. *University of Sydney Working Papers in Linguistics*, 2:1–59, 1981.

[37] Kathleen F. McCoy. Correcting misconceptions: What to say when the user is mistaken. In *Proceedings of the CHI'83; Conference on Human Factors in Computing Systems*, Boston, Ma. 1983.

[38] Kathleen R. McKeown. *Text Generation: Using Discourse Strategies and Focus Constraints to Generate Natural Language Text*. Cambridge University Press, Cambridge, England, 1985.

[39] Kathleen R. McKeown. Generating goal-oriented explanations. *International Journal of Expert Systems*, 1(4):377–395, 1988.

[40] Kathleen R. McKeown, Michael Wish, and Kevin Matthews. Tailoring explanations for the user. In *Proceedings of IJCAI-85*, Los Angeles, Ca., 1985. International Joint Conference on Artificial Intelligence.

[41] Marie Meteer. Defining a vocabulary for text planning, August 1988. Presented at the AAAI-88 Workshop on Text Planning and Realization.

[42] M.W. Meteer, D.D. McDonald, S.D. Anderson, D. Forster, L.S. Gay, A.K. Huettner, and P. Sibun. Mumble-86: Design and implementation. Technical Report 87-87, COINS, University of Massachusetts, 1987.

[43] Johanna D. Moore and William R. Swartout. A Reactive Approach to Explanation. In *Proceedings of the Eleventh International Joint Conference on Artificial Intelligence*, Detroit, MI, August 20-25 1989.

[44] Katharina Morik. Modeling the user's wants. Presented at the First International Workshop on User Modeling, Maria Laach, West Germany, August 1986.

[45] Robert Neches, William R. Swartout, and Johanna D. Moore. Enhanced Maintenance and Explanation of Expert Systems Through Explicit Models of Their Development. *IEEE Transactions on Software Engineering*, SE-11(11):1337–1351, November 1985.

[46] Hans-Joachim Novak. Generating referring phrases in a dynamic environment. In Michael Zock and Gérard Sabeh, editors, *Advances in Natural Language Generation: An interdisciplinary perspective; Volume 2*, pages 76–85. Pinter Publishers, London, 1988.

[47] Cécile L. Paris. *The Use of Explicit User Models in Text Generation: Tailoring to a User's Level of Expertise.* PhD thesis, Columbia University Department of Computer Science, 1987. To be published in the "Communication in Artificial Intelligence" series, Steiner and Fawcett (eds), Frances Pinter, 1990.

[48] Cécile L. Paris. Planning a text: can we and how should we modularize this process?, August 1988. Presented at the AAAI-88 Workshop on Text Planning and Realization.

[49] Cécile L. Paris. Tailoring Object Descriptions to the User's Level of Expertise. *Computational Linguistics*, 14 (3):64–78, September 1988. Special Issue on User Modeling.

[50] Terry Patten. Compiling the interface between text planning and realization, August 1988. Presented at the AAAI-88 Workshop on Text Planning and Realization.

[51] Rachel Reichman. *Plain Speaking: a Theory and Grammar of Spontaneous Discourse.* PhD thesis, Harvard University, May 1981.

[52] Norbert Reithinger. Generating referring expressions and pointing gestures. In G. Kempen, editor, *Natural Language Generation: Recent Advances in Artificial Intelligence, Psychology, and Linguistics*, pages 71–81. Kluwer Academic Publishers, Boston/Dordrecht, 1987. Paper presented at the Third International Workshop on Natural Language Generation, August 1986, Nijmegen, The Netherlands.

[53] Elaine A. Rich. User Modeling Via Stereotypes. *Cognitive Science*, 3:329 – 354, 1979.

[54] David Sanford and John W. Roach. Using the theory of metacommunication to guide natural language generation, August 1988. Presented at the AAAI-88 Workshop on Text Planning and Realization.

[55] David L. Sanford and John W. Roach. Parsing and generating the pragmatics of natural language utterances using metacommunication. In *Proceedings of the 9th Annual Conference of the Cognitive Science Society*, pages 89–95, 1987.

[56] David L. Sanford and John W. Roach. Using the theory of metacommunication to guide natural language generation, August 1988. Presented at the AAAI-88 Workshop on Text Planning and Realization.

[57] C.L. Sidner. Focusing for interpretation of pronouns. *American Journal of Computational Linguistics*, 7(4):217–231, 1981.

[58] *Stedman's Medical Dictionary*, Baltimore, London, Los Angeles, Sydney, 1982. Williams and Wilkins. 24th Edition.

[59] William R. Swartout and Stephen W. Smoliar. On making expert systems more like experts. *Expert Systems*, 4(3), August 1987.

[60] Jean Ure. Practical registers. *English Language Teaching Journal*, 23(2 and 3), 1969.

[61] Jean Ure and J. Ellis. Language varieties - register. In Meetham et al., editor, *Encyclopedia of Linguistics (Information and Control)*, volume 12, pages 251-259. Pergammon Press, Oxford, 1969.

[62] Peter van Beek. A model for generating better explanations. In *Proceedings of the 25th Annual Meeting of the ACL*, Palo Alto, California, 1987. Association of Computational Linguistics.

[63] *Webster's New Twentieth Century Dictionary*, New York, 1979. New World Dictionaries. Second Edition.

[64] R. White. Communicative competence, registers, and second language. *International Review of Applied Linguistics*, 12(2):127-141, 1974.

[65] Ursula Wolz. The impact of user modeling on text generation in task-oriented settings. In *Proceedings of the Second International Workshop on User Modeling*. Sponsored by AAAI and the University of Hawaii, 1990.